



Repository Service

Transcripción

[00:00] Listo, tenemos migración y tenemos entidad. ¿Qué es lo que necesitamos ahora? Si queremos traer objetos desde la base de datos, en este caso quiero traer usuarios de la base datos necesito un repositorio. Entonces voy a entrar aquí a usuarios y voy a crear una nueva interfaz llamada usuariosRepository.

[00:29] Tengo ya mi interfaz aquí de repositorio de usuarios. Lo dejo aquí. No lo agrego a Git aún y aquí lo que voy a hacer es extender de Repository y como primer parámetro tengo que decirle que es usuario, que es el tipo de objeto que va a manejar, y el tipo de dato de la llave primaria exactamente igual que en el caso de médicos.

[00:59] Fue mi error, no es CrudRepository, es JpaRepository. Eso lo arreglamos en este momento. Ambas interfaces son muy similares, tienden aplicaciones similares pero JpaRepository tiene métodos un poco más óptimos. No se preocupen. Bueno, ya tengo mi interfaz nuevamente, pues en plural, creo que me está completando el computador, lo que voy a hacer es renombrarlo a singular nada más. Y listo.

[01:30] Perfecto, UsuarioRepository extendiendo de JpaRepository y usuario long. Perfecto. Nada más. ¿Qué más necesito aquí? Aquí necesito ahora una capa intermedia que es mi capa de service, porque yo voy ejecutar cierta lógica en ese servicio para que Spring tome ese servicio y busque aquí en mi repositorio por los usuarios. Para eso aquí en infra yo tengo mi tratador de errores.

[02:00] Lo que yo quiero hacer aquí ahora es crear un servicio de autenticación de usuarios. Pero no quiero que se mezcle con mi tratador de errores, entonces lo que voy a hacer aquí es crear un nuevo paquete llamado errores, este va a ir aquí y voy a crear otro paquete llamado Security. Entonces mi tratador de errores ahora va a ir aquí en este paquete, voy a refactorizar y en Security yo voy a crear una nueva clase llamada AutenticaciónService.

[02:42] Esta clase es la que va a tener la lógica de autenticación dentro de mi aplicación. Ahora como siempre, si yo deseo decirle a Spring: “por favor Spring escanea esta clase, porque es un servicio que yo quiero abastecer para mi aplicación”, ¿qué es lo que tengo que hacer? Exacto, tengo que marcarla con el estereotipo service.

[03:06] Este estereotipo va a decirle a Spring: “Spring, escanea esta clase porque es un servicio de esta aplicación”. ¿Qué más necesito hacer ahora? Ahora es lo que tengo que hacer es extender, acá comienza ya la magia. Perdón, extender no. Es implementar porque es una interfaz, puede haber implement, de una interfaz propia de Spring que Spring utiliza para efectuar su login.

[03:29] Que es nada más y nada menos que el UserDetailsService. Ahora vemos que marqué en rojo porque me pide que implemente los métodos de esa interfaz, pero ya dense cuenta como por ejemplo ya con esta clase nosotros implementamos una interfaz propia de Spring que él utiliza internamente, está internamente cableado digamos para lo que es el proceso de autenticación del usuario.

[04:00] Voy a implementar los métodos, voy a dar okay, y listo, aquí tengo un objeto de tipo UserDetails y me dice loadUserByUsername. Significa cargar usuario por nombre de usuario. Aquí lo que me está pidiendo es, en este método lo que yo le tengo que decir es de qué forma yo voy a cargar ese usuario y de dónde, porque a Spring no le interesa el datasource o la fuente de datos de la cual yo obtengo el usuario.

[04:30] A Spring le interesa, que yo le dé el usuario, dependiendo del username que estaba buscando. ¿Esto a qué les suena? Exacto. A un método que yo tengo que crear aquí en mi repositorio, entonces para esto, lo que yo tengo que hacer primero es venir aquí y declarar mi repositorio.

[04:50] Entonces, voy a decirle UsuarioRepository usuarioRepository; recuerden para inyectarlo con @Autowired, yo lo voy a hacer aquí a nivel del campo, pero recuerden que para métodos de testing esto no es recomendado, pero como no es el objetivo del curso enfocarnos ahí, vamos a dejarlo en @Autowired. Puede ser un setter o puede ser un constructor, no se preocupen.

[05:13] Y aquí le voy a decir que retorne usuarioRepository.findBy y lo tengo Username. Le voy a decir Username. Y como parámetro el Username. Me está dando rojo porque no existe. ¿Lo que hago ahora qué es? Crearlo. Y listo. Vemos que con un objeto del tipo UserDetails me va a encontrar mi usuario por nombre de usuario en mi interfaz usuarioRepository.

[05:48] Terminamos este segundo paso. Entonces recapitulando, tenemos la entidad usuario, tenemos la migración que crea la tabla para mis usuarios en mi base de datos. Y ahora yo tengo el repositorio con mi servicio de autenticación que implementa esta interfaz propia de Spring userDetailsService para comenzar a decirle a Spring que yo quiero personalizar el flujo de autenticación que tiene por defecto. Nos vemos en el siguiente video.